Course Title: **Compilers and Languages** اللغات والمترجمات Course Code: CCE3113 3rd year
Date: 15.1.2017 (First term) Allowed time: 3 hrs No. of Pages: (4)

Answer the following five questions:

Question No. 1 **(25 Marks)**

1. Write down a recursive descent parser (i.e. parsing algorithm) for the following grammar: (5 marks)

   $E \rightarrow T \mid T + E$

   $T \rightarrow int \mid int * T \mid (E)$

2. Consider the following CFG (20 Marks- 5 marks for each point)

   $S \rightarrow A$

   $A \rightarrow BC \mid DBC$

   $B \rightarrow Bb \mid \varepsilon$

   $C \rightarrow c \mid \varepsilon$

   $D \rightarrow a \mid d$

   a) Is this grammar suitable to be parsed using the recursive descendent parsing method? Explain your answer and modify the grammar if needed.

   b) Compute the FIRST and FOLLOW set of non- terminal symbols of the grammar resulting from your answer in a)

   c) Construct the corresponding parsing table using the predictive parsing LL method.

   d) Show the stack contents, the input and the rules used during parsing for the input w = dbb

Question No. 2 **(20 Marks)**

1. Given the following context free grammar (5 Marks)

   $S \longrightarrow bAb \mid bBa$

   $A \longrightarrow aS \mid CB$

   $B \longrightarrow b \mid BC$

   $C \longrightarrow c \mid cC$

   Explain with example why the grammar is ambiguous

2. Consider the regular expression below which can be used as part of a specification of the definition of exponents in floating-point numbers. Assume that the alphabet consists of numeric digits ('0' through '9') and alphanumeric characters ('a' through 'z' and 'A' through 'Z') with the addition of a selected small set of punctuation and special characters (say in this example only the characters '+' and '-' are relevant). Also, in this representation of regular expressions the character '.' denotes concatenation. For the following regular expression

   Exponent = (+ | - | ε) . (E | e) . (digit)+

   Derive the DFA that is able to recognize this language. (5 Marks)

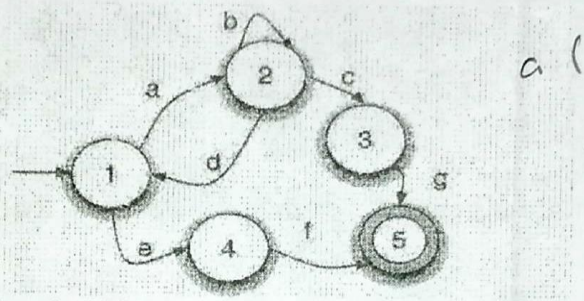
3. Compare between LL parser and LR parser. (Discuss at least five comparison points). (5 Marks)

4. List the main phases of a compiler illustrating the job performed by each. What distinguishes the front end of a compiler from the back end? (5 Marks)

Question No. 3 **(20 Marks)**

For each of the following, please choose the letter introducing the best answer. Explain your answer. (Each one is worth three degrees.....Note that: the explanation is worth one degree.)

1. How many strings of length less than or equal 3 are in the language described by the regular expression $(x + y)^* y (a + ab)^*$?
   a) 7
   b) 10
   c) 12
   d) 11
2. Which of the following regular expressions generate the same language as the one recognized by this NFA?



   a) (ab + d)* ((ab*cg) + (e f))
   b) (ab*d)* ((ab + cg) + (e f))
   c) (ab*d)* +((ab*cg) + (e f))
   d) (ab*d)* ((ab*cg) + (e f))
3. Part of the compiler that takes as input a stream of characters and produces as output a stream of words along with their associated syntactic categories.
   a) Scanner
   b) Parser
   c) Front end
   d) Code generator
4. How many strings does the following grammar generate?

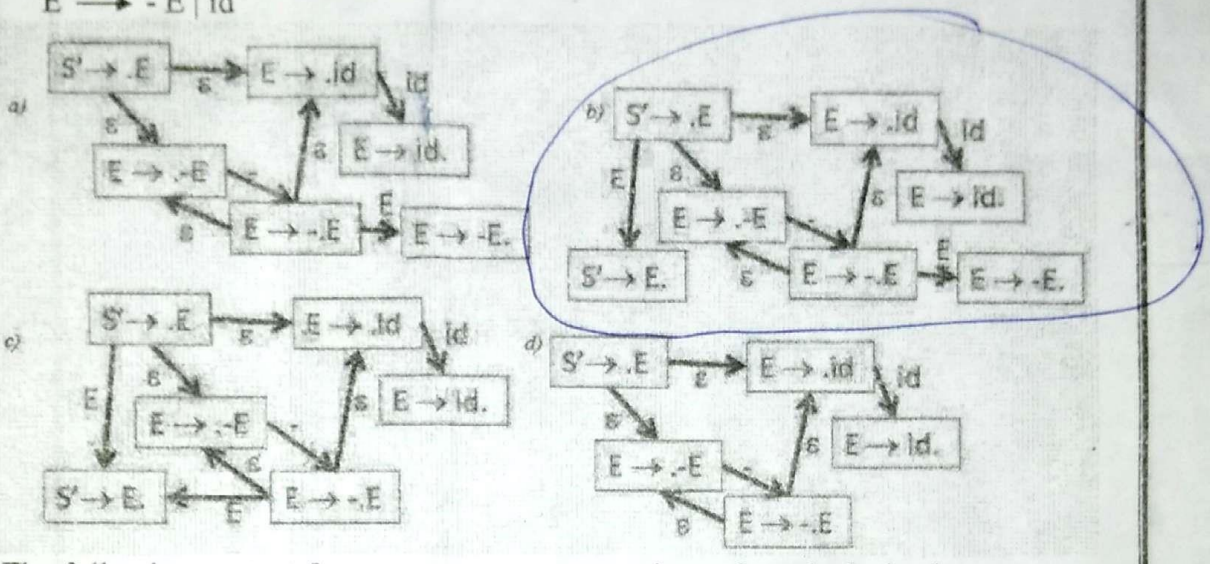
   $A \rightarrow BB$
   $B \rightarrow CC$
   $C \rightarrow 1 \mid 2$

   a) 8
   b) 12
   c) 16
   d) 32
5. Choose the grammar that correctly eliminates left recursion from the given grammar: $S \rightarrow S + a \mid S + b \mid c$
   a) $S \rightarrow S + L \mid c$
      $L \rightarrow a \mid b$
   b) $S \rightarrow c\,L$
      $L \rightarrow + a\,L \mid + b\,L \mid \in$
   c) $S \rightarrow S\,L$
      $L \rightarrow + a \mid + b \mid \in$
   d) $S \rightarrow S\,L$
      $L \rightarrow + a \mid + b$

6. Choose the correct NFA for the given grammar?

S' ⟶ E

E ⟶ - E | id



7. The following context free grammar generates strings of terminals that have:

S ⟶ aB | bA

A ⟶ a | aS | bAA

B ⟶ b | bS | aBB

a) equal number of a's and b's

b) odd number of a's and odd number b's

c) even number of a's and even number of b's

d) odd number of a's and even number of a's

8. Which of the following pairs of regular expressions are equivalent?

a) 1 (01)* and (10)* 1

b) x (xx) * and (xx) * x

c) x^+ and x^+x^{*+}

d) All of these

9. Choose the expression that the given assembly code was generated from.

a) 5 + (4 - 3)

b) 5 - (4 + 3)

c) (5 + 4) - 3

d) (5 - 4) + 3

```
li $a0 5
sw $a0 0($sp)
addiu $sp $sp -4
li $a0 4
sw $a0 0($sp)
addiu $sp $sp -4
li $a0 3
lw $t1 4($sp)
sub $a0 $t1 $a0
addiu $sp $sp 4
lw $t1 4($sp)
add $a0 $t1 $a0
addiu $sp $sp 4
```

10. A bottom up parser generates

a) Right most derivation

b) Right most derivation in reverse

c) Left most derivation

d) Left most derivation in reverse

Question No. 4 **(15 Marks)**

1. Consider the following CFG:

   S → A
   S → xb
   A → aAb
   A → B
   B → x

   (a) Construct a DFA for viable prefixes of this grammar using LR(0) items. Show the contents of each state (set of items) and label the transitions clearly. (5 marks)

   (b) Is this grammar SLR(1)? Briefly explain why or why not. (5 marks)

2. Consider the following shift-reduce parse of a string:

   |aabaa$ ->
   a|abaa$ ->
   aa|baa$ ->
   aab|aa$ ->
   aaS|aa$ ->
   aaSa|a$ ->
   aS|a$ ->
   aSa|$ ->
   S

   Assuming this parse exercises all productions of the grammar, what is the grammar for this shift-reduce parser? (5 marks)

Question No. 5 **(20 Marks)**

1. Consider the following basic block, in which all variables are integers.

   a := 0 * f
   b := f * v
   c := b + b
   d := a * 3
   x := f * v
   y := x + x
   z := y / c

   Assume that the only variables that are live at the exit of this block are y and z, while v and f are given as inputs. In order, apply the following optimizations to this basic block. Show the result of each transformation. For each optimization, you must continue to apply it until no further transformations are possible, before writing out the result and moving on to the next.

   (Note that: each optimization step is worth 2 marks)

   (a) Algebraic simplification
   (b) Copy propagation
   (c) Common sub-expression elimination
   (d) Constant folding
   (e) Copy propagation
   (f) Dead code elimination

   When you have completed the last of the above transformations, the resulting program will still not be optimal. What optimizations, in what order, can you apply to optimize the result further? (3 marks)

2. Left factor the following grammar: S → a S c | a S b | b (5 marks)

Best wishes
Dr. Sherin El Gokhy